|  |
| --- |
| Unit 4: Programming |
| Abdulloh Halilov |

**O‘qituvchi:** Ilyoskhuja Ikromkhujaev

**Guruh identifikatori:** 23-412

**Talaba ID:** 231397

**Taqdim etilgan sana:** 20.01.2025 yil

**BTEC o'quvchilar topshiriqlarini baholash va deklaratsiya**

Baholash uchun ishlarni taqdim etganda, har bir o'quvchi ish o'ziniki ekanligini tasdiqlovchi deklaratsiyani imzolashi kerak.

|  |  |
| --- | --- |
| **O‘quvchi (talaba) identifikatori:** | 231397 |
| **Baholovchi nomi:** | Ilyoskhuja Ikromkhujaev |
| **BTEC dasturi nomi:** | Pearson BTEC Higher Nationals in Digital Technologies |
| **Birlik yoki komponent raqami va nomi:** | Unit 4: Programming |
| **Topshiriq nomi:** | Internal Assignment |
| **Topshiriq topshirilgan sana:** | 20.01.2025 yil |

Iltimos, har bir topshiriq uchun berilgan ishlarni sanab o'ting. Ishlarni topish mumkin bo'lgan sahifa raqamlarini ko'rsating yoki ishlarning mohiyatini tavsiflang (masalan, grafik, rasm).

|  |  |  |
| --- | --- | --- |
| **Topshiriq vazifasi ma'lumoti** | **Ishlar taqdim etildi** | **Sahifa** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
| **O'quvchi deklaratsiyasi**  Men ushbu topshiriq uchun taqdim etilgan ish meniki ekanligini tasdiqlayman. Ishda foydalanilgan manbalarga aniq havola qildim. Men noto'g'ri deklaratsiya noto'g'ri ishlashning bir shakli ekanligini tushunaman.  **O‘quvchi imzosi:**  **Sana:** 20.01.2025 yil |

Mundarija

[Kirish 3](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462352)

[1-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462353) 5

[2-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462354) ?

[3-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462355) ?

[4-qism](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462356) ?

[Adabiyotlar ro’yxati](file:///C:\Users\Lochinbek\Desktop\PDP\BigData\Assignment\yangisi\Big%20Data%20and%20BA%20Ass.Template%20EN.docx#_Toc134462358) ?

KIRISH

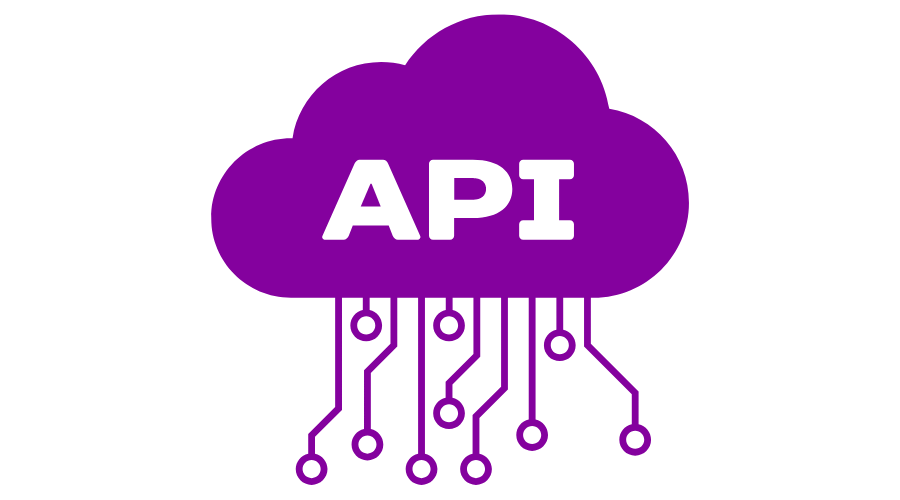
Assalomu alaykum. NewEra Cash & Carry uchun ASP.NET API-ni ishlab chiqish va hujjatlashtirish bo'yicha topshirig'imga xush kelibsiz. Ushbu loyiha modulimizning o'quv maqsadlariga erishish uchun amalga oshirialdi. Mening maqsadim nafaqat ushbu tushunchalarni tushunishimni namoyish etish, balki ularni ASP.NET API ishlab chiqish doirasida amalda qo'llashdir. Ushbu kirish bo'limi nazariy bilimlarni qanday qilib funktsional dasturiy echimlarga aylantirish mumkinligini chuqurroq o'rganishga yordam beradi.

1-QISM

**LO1**

**A.P1**

API so'rovlarni ko'rib chiqish va tegishli javoblarni taqdim etish uchun aniq jarayonni kuzatish orqali ishlaydi. So'rov yuborilganda, API birinchi navbatda uning to'g'ri qabul qilinishini ta'minlaydi va so'rovning URL manzili va HTTP usuli (GET, POST, PUT yoki DELETE kabi) asosida mo'ljallangan oxirgi nuqtani aniqlaydi.



Keyinchalik, JWT Bearer tokeni bilan autentifikatsiya hisob maʼlumotlarini tekshirish orqali soʻrov vakolatli manbadan ekanligini tekshiradi. Shuningdek, foydalanuvchi so'ralgan manbaga kirish uchun zarur ruxsatlarga ega ekanligini tekshiradi. So'rov ushbu xavfsizlik tekshiruvlaridan o'tgandan so'ng, API taqdim etilgan ma'lumotlarni tekshdiradi va barcha kerakli maydonlar mavjudligini va to'g'ri formatlanganligini ta'minlaydi.

Tekshirishdan so'ng, API so'rovni tegishli boshqaruvchiga va harakat usuliga yo'naltirish orqali qayta ishlaydi. So'rovning maqsadiga qarab, bu bosqich ma'lumotlar bazasi yoki boshqa ma'lumotlar manbalari bilan o'zaro aloqani o'z ichiga olishi mumkin. Misol uchun, Admin API so'rovda taqdim etilgan Product ma'lumotlari asosida yozuvlarni olishi, yasashi, yangilashi yoki o'chirishi mumkin. Agar customer bo’lsa faqat Mahsulot ma’lumotlarini ko’ra oladi.

Kerakli operatsiyalar bajarilgandan so'ng, MVC mijoz uchun javob tayyorlaydi. Bu javob JSON kabi tuzilgan formatdagi maʼlumotlarni oʻz ichiga oladi va soʻrov muvaffaqiyatli yoki xatolik yuz berganligini koʻrsatish uchun HTTP holat kodini oʻz ichiga oladi (masalan, muvaffaqiyat uchun 200, notoʻgʻri soʻrov uchun 400, ruxsatsiz kirish uchun 401 va hokazo). .

Nihoyat, API javobni mijozga yuboradi va so'rov va javob tafsilotlarini qayd qiladi. Ushbu Request Response recordlari Middleware yordamida fayllarga saqlanadi. Ushbu jarayon API so'rovlarni samarali va xavfsiz tarzda ko'rib chiqishni ta'minlaydi.

**A.M1**

**(**ASP.NET Core'ni ishda ishlayotgan til tarjimoni deb tasavvur qiling, lekin bu kompyuterlar uchun. C# tilida kod yozganingizda, bu biz odamlar uchun o'qilishi mumkin bo'lgan ingliz tilidagi ko'rsatmalarni yozish kabi bo'ladi, lekin kompyuterlar bevosita shu tilda gaplashmaydi.

Birinchi navbatda, C# kodingiz Roslyn deb nomlangan tarjimonga o'tadi. Roslyn sizning odamlar uchun qulay bo'lgan kodingizni Intermediate Language (IL) deb nomlangan narsaga aylantiradi. Bu IL — har qanday kompyuter to'g'ri yordam bilan tushunishi mumkin bo'lgan universal tilga o'xshaydi.

Keyin esa juda aqlli qism boshlanadi — .NET Core runtime (CoreCLR) ishga tushadi. Bu sizning kompyuteringiz gaplashadigan aniq dialektga (mashina kodi) IL-ni tarjima qiladigan shaxsiy tarjimoningizga o'xshaydi. Bu jarayon Just-In-Time (JIT) kompilatsiyasi deb ataladi.

Kimdir veb-saytingizga tashrif buyurganda, so'rov Kestrel orqali keladi (buni veb-saytingizning qabulxonasi deb tasavvur qiling). Endi kodingiz, kompyuterning mahalliy tilida gaplashayotgan bo'lsa, so'rovni qayta ishlash uchun ishga tushadi. Runtime barcha orqa fondagi ishlarni boshqaradi — ishlatilmayotgan xotiralarni tozalash, bir nechta vazifalarni bir vaqtning o'zida bajarish va hamma narsa to'g'ri ishlashini ta'minlash.

Eng aqlli qismi shundaki, kodingiz vaqt o'tishi bilan yaxshilanadi. Eng ko'p ishlatiladigan qismlar qo'shimcha optimallashtirishni oladi — bu siz yangi tilni o'rganayotganda tez-tez ishlatadigan iboralarni aytishda yaxshilanishingizga o'xshaydi. Runtime bu optimallashtirilgan tarjimalarni eslab qoladi, bu esa dasturiy ta'minotingizni tezroq va samaraliroq ishlashini ta'minlaydi.

)

**Kodni yozishdan bajarishgacha bo'lgan jarayon bir nechta muhim bosqichlarni o'z ichiga oladi, bu dastur funktsiyalarini mo'ljallangan tarzda ta'minlaydi. Bu kod yozishdan boshlanadi, bu yerda ishlab chiquvchi algoritm yoki mantiqni ASP.NET uchun C# kabi dasturlash tiliga tarjima qiladi. Ushbu bosqichda eng yaxshi amaliyotlarga rioya qilish, jumladan toza, modulli va yaxshi hujjatlashtirilgan kodni saqlash juda muhimdir.**

**Kod yozilgach, u kompilyatsiya jarayonidan o'tadi. Kompilyatsiya - bu kod inson tomonidan o'qiladigan yuqori darajadagi sintaksisdan oraliq yoki mashinada o'qiladigan kodga aylantiriladi. Masalan, ASP.NET da yozma kod .NET ish vaqti sharhlashi mumkin bo'lgan oraliq tilga (IL) kompilyatsiya qilinadi. Ushbu bosqich sintaktik xatolarni hal qilishni o'z ichiga oladi va kodning sintaktik jihatdan to'g'riligini va bajarishga tayyorligini ta'minlaydi.**

**Muvaffaqiyatli kompilyatsiyadan so'ng, dastur qurish bosqichiga o'tadi, bu erda loyihaning barcha komponentlari, masalan, kutubxonalar va resurslar joylashtiriladigan birlikka to'planadi. Bu ko'pincha server ishga tushirishi mumkin bo'lgan bajariladigan fayllarni (masalan, .dll yoki .exe) yaratishga olib keladi.**

**Qurilishdan so'ng dastur sinov bosqichiga o'tadi. Ushbu bosqichda kod kutilgandek harakat qilishini tekshirish uchun boshqariladigan muhitda bajariladi. Bu har qanday mantiqiy yoki ish vaqtidagi xatolarni aniqlash va hal qilish uchun birlik testi, integratsiya testi va funktsional testni o'z ichiga olishi mumkin.**

**Ilova sinovdan o'tgan va xatolar bartaraf etilgandan so'ng, u o'rnatishga tayyor. Joylashtirish ilovani maqsadli muhitga, masalan, ishlab chiqarish serveriga ko'chirishni o'z ichiga oladi, bu erda u oxirgi foydalanuvchilar uchun ochiq bo'ladi. ASP.NET API bo'lsa, o'rnatish APIni IIS kabi veb-serverda yoki Azure kabi bulut xizmatida joylashtirishni o'z ichiga olishi mumkin.**

**Joylashtirilgandan so'ng, dastur ishlashga tayyor. Foydalanuvchilar yoki mijoz tizimlari so'rovlarni yuborish orqali dastur bilan o'zaro aloqada bo'lib, server o'rnatilgan mantiq asosida qayta ishlaydi. Amalga oshirish jarayonida dastur xotira, mavzular va boshqa resurslarni samarali boshqarish uchun .NET Core ish vaqti kabi ish vaqti muhitlariga tayanadi.**

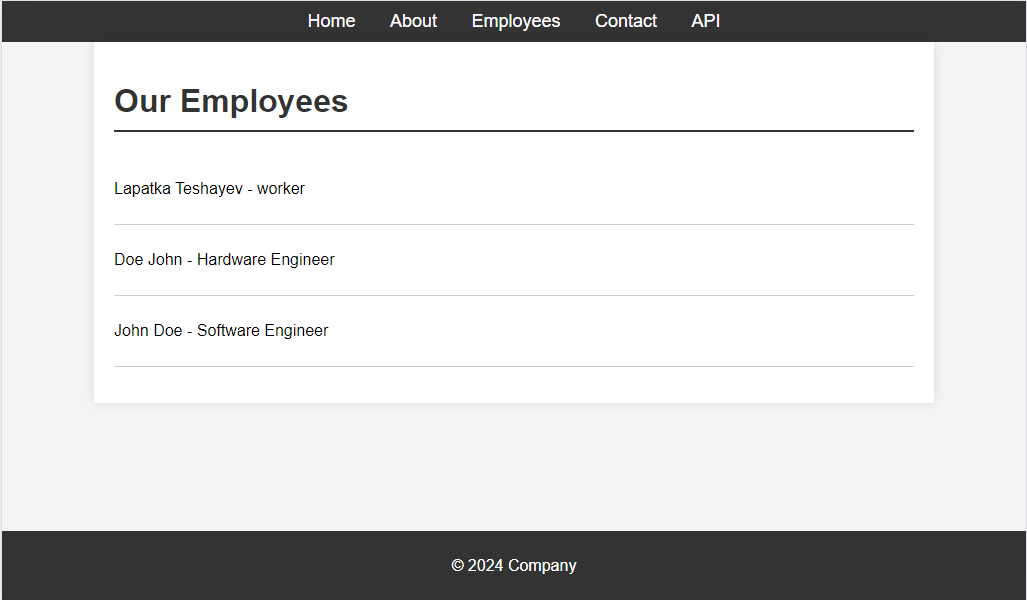
**Butun jarayon takrorlanadi, chunki ishlab chiquvchilar funksiyalar qo‘shish, xatolarni tuzatish yoki ish faoliyatini optimallashtirish uchun kod yozish va o‘zgartirishga qaytishi mumkin, bu esa ilova foydalanuvchi ehtiyojlari va tizim talablariga javob berishda davom etishini ta’minlashi mumkin.**

**A.D1**

**Veb saytning maqsadi va foydalanuvchining ehtiyojlari uning ishlab chiqish jarayoniga quyidagicha ta'sir qiladi**

**Maqsad va foydalanuvchining ehtiyojlarini aniqlash**

**Saytning asosiy maqsadi kompaniya haqida to'liq va yangilanadigan ma'lumotlar berishdir. Foydalanuvchilarning asosiy ehtiyojlari kompaniya haqida ma'lumot olish kompaniya xodimlari haqida batafsil ma'lumotlar olish va kompaniya bilan bog'lanishdir. Ushbu ehtiyojlar ishlab chiqish jarayonida asosiy e'tiborni foydalanuvchilar uchun qulay interfeys va to'liq ma'lumotlar taqdim etishga qaratadi**



**Django frameworkining tanladik**

**Saytning dinamik va yangilanadigan ma'lumotlarni taqdim etish talabi Django frameworkini tanlashga olib keldi. Django kuchli admin panel va modullari orqali ma'lumotlarni boshqarish imkonini beradi. Bu maqsad va ehtiyojlarga mos ravishda ma'lumotlarni oson boshqarish va yangilashni ta'minlaydi**



**HTML va CSS yordamida frontend dizayn**

**Foydalanuvchilarning qulay interfeys va chiroyli dizayn talablarini qondirish uchun HTML va CSS texnologiyalari tanlandi. Bu texnologiyalar foydalanuvchilarga tushunarli va estetik jihatdan yoqimli sayt yaratish imkonini beradi. Sahifalar tuzilishi oddiy va aniq bo'lishi kerakligi ishlab chiqish jarayonida HTML va CSS ni qo'llashni talab qiladi**

**Ma'lumotlar bazasi tanlovi**

**Saytning maqsadi va foydalanuvchi ehtiyojlariga ko'ra, ma'lumotlar bazasi sifatida SQLite yoki PostgreSQL tanlanadi. Bu tanlov ma'lumotlarni ishonchli saqlash va boshqarishni ta'minlaydi. Bu foydalanuvchilarning yangilanadigan ma'lumotlarga tez va oson kirish imkoniyatini yaratadi**

**Admin panel orqali ma'lumotlarni boshqarish**

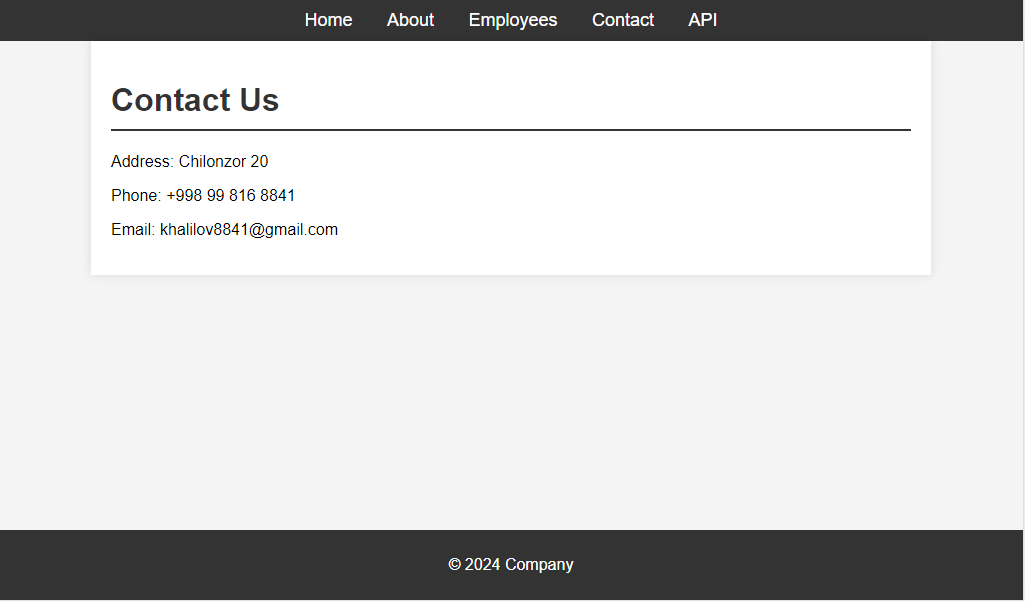
**Sayt maqsadi va foydalanuvchi ehtiyojlariga ko'ra, admin panel orqali ma'lumotlarni kiritish va yangilash imkoniyati yaratiladi. Bu admin uchun qulay va oson boshqarish imkonini beradi, shuningdek saytni doimiy ravishda yangilab turishga yordam beradi**

**umuman olganda, veb saytning maqsadi va foydalanuvchining ehtiyojlari ishlab chiqish jarayoniga katta ta'sir qiladi. Bu talablarga mos ravishda texnologiyalar tanlanadi, dizayn va funksiyalar ishlab chiqiladi, ma'lumotlar bazasi va admin panel yaratilib, foydalanuvchilar uchun qulay va samarali sayt ta'minlanadi**

**A.D1**

**Veb saytning maqsadi va foydalanuvchining ehtiyojlari uning ishlab chiqish samaradorligiga sezilarli ta'sir ko'rsatadi. Quyida bu ta'sirlarni batafsil baholab chiqamiz.**

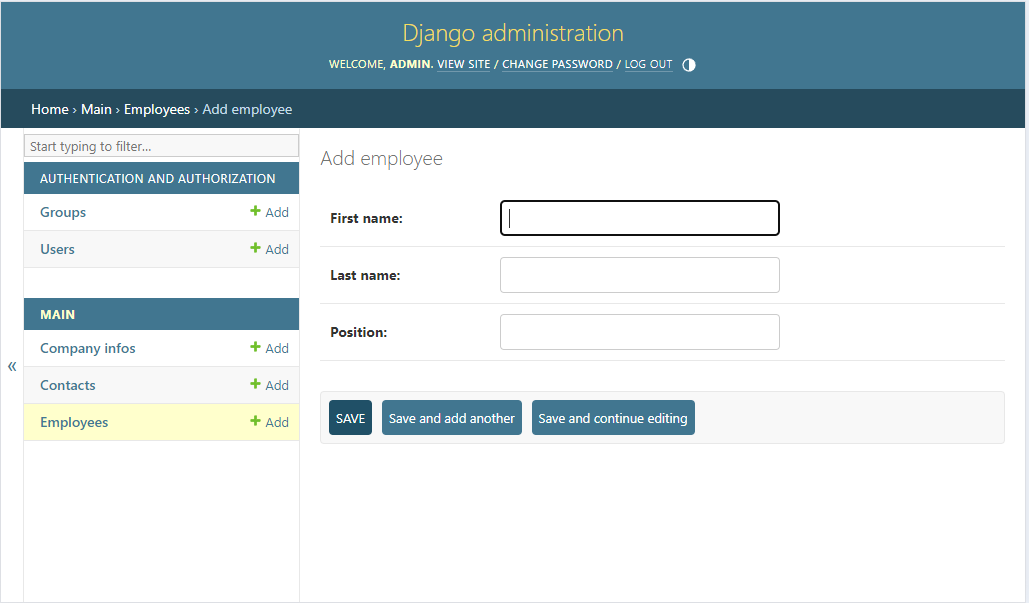
**Veb saytning asosiy maqsadi kompaniya haqida to'liq va yangilanadigan ma'lumotlarni taqdim etishdir. Foydalanuvchilarning asosiy ehtiyojlari kompaniya haqida ma'lumot olish, kompaniya xodimlari haqida batafsil ma'lumotlarni ko'rish va kompaniya bilan bog'lanishdir. Ushbu ehtiyojlar ishlab chiqish jarayonida aniq yo'nalish beradi va ortiqcha vaqt va resurslar sarflashning oldini oladi.**



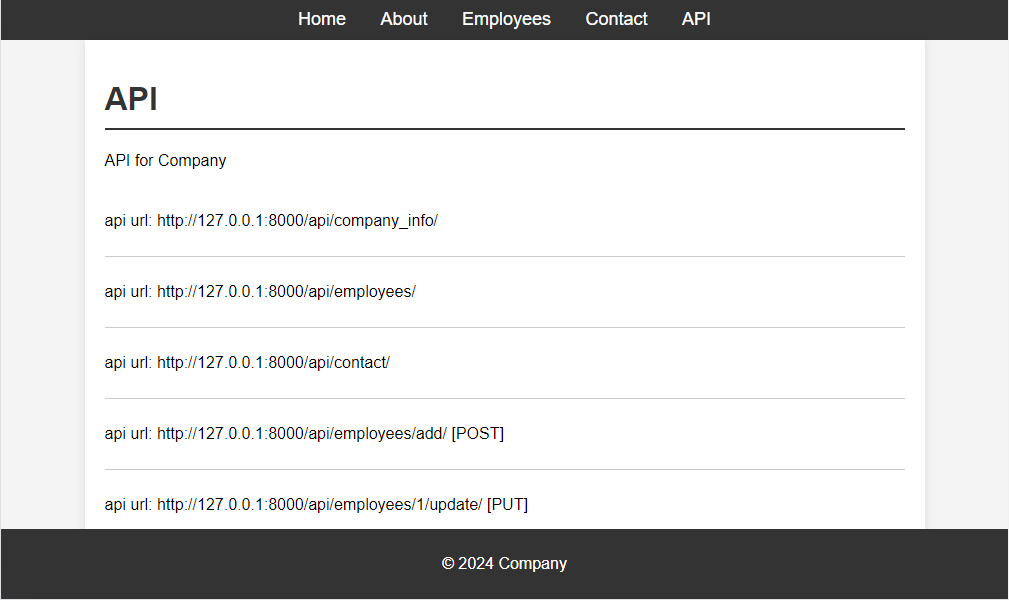
**Maqsad va ehtiyojlarga mos ravishda Django frameworkining tanlanishi ishlab chiqish samaradorligini oshiradi. Django kuchli admin paneli va modullar bilan ishlash imkoniyati ma'lumotlarni boshqarishni osonlashtiradi va tezlashtiradi. HTML va CSS yordamida qulay va chiroyli dizayn yaratish imkoniyati foydalanuvchilarga qulay interfeys yaratish imkonini beradi.**

**Foydalanuvchilarning qulay interfeys va tezkor sahifalar talabi ishlab chiqish jarayonida asosiy e'tibor qaratiladigan jihat bo'ladi. Bu talablar dizayn va funksiyalarni foydalanuvchi tajribasini yaxshilashga qaratishga yordam beradi. Sahifalar tez yuklanishi va mobil qurilmalarda yaxshi ko'rinishi uchun responsiv dizayn qo'llaniladi. Bu esa ishlab chiqish samaradorligini oshiradi va foydalanuvchilarning saytdan foydalanish tajribasini yaxshilaydi.**

**Admin panel orqali ma'lumotlarni boshqarish imkoniyati ishlab chiqish samaradorligini sezilarli darajada oshiradi. Bu orqali adminlar ma'lumotlarni osongina kiritish va yangilash imkoniyatiga ega bo'ladilar. Bu esa saytning doimiy ravishda yangilanib turishini ta'minlaydi va foydalanuvchilarga har doim yangilanib turadigan ma'lumotlar taqdim etadi.**



**API qo'shilishi esa ma'lumotlarni qo'shish, o'zgartirish, ko'rish va qidirish imkoniyatlarini beradi, bu esa foydalanuvchilarga va dasturchilarga qulaylik yaratadi. API orqali ma'lumotlarni boshqarish va olish imkoniyati, veb saytning samaradorligini oshiradi, chunki bu turli platformalar va qurilmalardan ma'lumotlar bilan ishlashni osonlashtiradi. API-ning mavjudligi, foydalanuvchilar va mijozlar uchun o'zaro aloqani soddalashtiradi va avtomatlashtirish jarayonlarini tezlashtiradi.**



**Umuman olganda, veb saytning maqsadi va foydalanuvchining ehtiyojlari ishlab chiqish samaradorligiga katta ta'sir ko'rsatadi. Aniq maqsad va ehtiyojlar ishlab chiqish jarayonini tezlashtiradi, texnologiyalarni samarali tanlashga yordam beradi, foydalanuvchi tajribasini yaxshilaydi va ma'lumotlarni boshqarishni osonlashtiradi. Bu esa saytni tez va samarali yaratishga olib keladi.**

XULOSA

Veb saytning maqsadi va foydalanuvchining ehtiyojlari uning ishlab chiqish samaradorligiga katta ta'sir ko'rsatadi. Aniq belgilangan maqsad va ehtiyojlar ishlab chiqish jarayonini samarali va tez bajarishga yordam beradi. Django frameworkining tanlanishi va HTML, CSS yordamida dizaynning yaratilishi jarayonni yanada osonlashtiradi. Foydalanuvchi tajribasini (UX) yaxshilashga qaratilgan qadamlar, sahifalarning tez yuklanishi va responsiv dizayn, saytdan foydalanishni qulaylashtiradi. Admin panel orqali ma'lumotlarni boshqarish imkoniyati esa saytni doimiy yangilanib turishini ta'minlaydi. API qo'shilishi esa ma'lumotlarni qo'shish, o'zgartirish, ko'rish va qidirish imkoniyatlarini beradi, bu esa foydalanuvchilarga va dasturchilarga qulaylik yaratadi. Umuman olganda, maqsad va ehtiyojlarga asoslangan ishlab chiqish jarayoni saytni tez va samarali yaratishga olib keladi.

adabiyotlar ro'yxati

1. Grinstein, G., and Plaisant, C., 2009. Interactive Information Visualization. Springer.
2. Nielsen, J., 1994. Usability Engineering. Academic Press.
3. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., and Diakopoulos, N., 2017. Designing the User Interface: Strategies for Effective Human-Computer Interaction. 6th ed. Pearson.
4. Krug, S., 2014. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. New Riders.
5. Cooper, A., Reimann, R., and Cronin, D., 2007. About Face 3: The Essentials of Interaction Design. 3rd ed. Wiley.
6. Garrett, J.J., 2010. The Elements of User Experience: User-Centered Design for the Web and Beyond. 2nd ed. New Riders.
7. Norman, D.A., 2013. The Design of Everyday Things: Revised and Expanded Edition. Basic Books.